Software Requirements Specification

for

JustRentIt

Version 1.0 approved

Prepared by –

Aanish Rahul Bangre (UID - 2023300011)

Arjun Bagade (UID - 2023300008)

August 28, 2025

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive and detailed description of the requirements for **"JustRent-It"**, a web-based peer-to-peer item rental platform. This document serves as the authoritative source for system requirements and will guide the development, testing, validation, and deployment phases of the project.

Primary Objectives:

- Define functional and nonfunctional requirements clearly
- Establish system boundaries and constraints
- Provide a foundation for project planning and resource allocation
- Serve as a reference for quality assurance and testing activities
- Enable effective communication among stakeholders

1.2 Document Conventions

This document follows IEEE 830-1998 standards for Software Requirements Specifications with the following conventions:

- **Requirements Labeling:**
  - REQ-X: General system requirements
  - FR-X: Functional requirements
  - NFR-X: Nonfunctional requirements
  - UI-X: User interface requirements
  - IF-X: Interface requirements
- **Priority Levels:**
  - **High**: Critical for system operation
  - **Medium**: Important for user satisfaction
  - **Low**: Desirable but not essential
- **Text Formatting:**
  - **Bold**: Key terms and important concepts
  - *Italics*: References to external documents
  - Code: Technical specifications and system components

1.3 Intended Audience and Reading Suggestions

This document is designed for multiple stakeholder groups:

Primary Audience:

- **Software Developers**: Implementation guidance, technical specifications, and system architecture details
- **Quality Assurance Team**: Test case development, validation criteria, and acceptance testing
- **Project Managers**: Scope definition, milestone tracking, and resource planning
- **Business Stakeholders**: Requirements validation and business value assessment

Secondary Audience:

- **System Administrators**: Deployment and maintenance requirements
- **End Users**: Understanding system capabilities and limitations
- **Regulatory Compliance Teams**: Security and privacy requirement validation

Reading Recommendations:

1. **Executive Summary**: Start with Section 1.4 (Product Scope)
2. **Technical Overview**: Proceed to Section 2 (Overall Description)
3. **Detailed Requirements**: Review Section 4 (System Features)
4. **Quality Attributes**: Study Section 5 (Nonfunctional Requirements)
5. **Implementation Details**: Reference Section 3 (External Interface Requirements)

1.4 Product Scope

JustRent-It is a comprehensive web-based platform designed to facilitate secure, efficient, and user-friendly peer-to-peer item rentals. The system addresses the growing sharing economy by providing individuals with a trusted platform to monetize underutilized assets while offering renters access to items without the burden of ownership.

Core Value Proposition:

- **For Item Owners**: Generate income from idle assets with minimal effort
- **For Renters**: Access to diverse items without long-term commitment or storage concerns
- **For the Community**: Promote sustainable consumption and resource sharing

System Boundaries:

- **In Scope**: User management, item cataloging, booking workflows, communication tools, payment processing integration points
- **Out of Scope**: Physical item delivery, insurance services, dispute resolution legal processes, third-party payment processing

Success Metrics:

- User registration and retention rates
- Item listing and booking conversion rates
- Platform transaction volume and revenue
- User satisfaction and safety ratings

1.5 References

Technical Documentation:

- *Next.js Documentation* - https://nextjs.org/docs
- *Appwrite Documentation* - https://appwrite.io/docs
- *Tailwind CSS Documentation* - https://tailwindcss.com/docs

Standards and Guidelines:

- IEEE 830-1998: Recommended Practice for Software Requirements Specifications
- WCAG 2.1: Web Content Accessibility Guidelines
- GDPR: General Data Protection Regulation
- OWASP Security Guidelines

---

2. Overall Description

2.1 Product Perspective

JustRent-It is a standalone, cloud-based web application that operates independently within the sharing economy ecosystem. The system leverages modern web technologies to provide a scalable, maintainable, and secure platform for peer-to-peer transactions.

System Architecture Overview:

- **Frontend Layer**: Next.js-based responsive web application with React components
- **Backend Layer**: Appwrite Backend-as-a-Service (BaaS) for authentication, database, and file storage
- **Integration Layer**: RESTful APIs for third-party service integration
- **Data Layer**: Cloud-hosted database with automated backup and recovery

Key Architectural Principles:

- **Modularity**: Component-based architecture enabling independent development and testing
- **Scalability**: Horizontal scaling capabilities to accommodate user growth
- **Security**: Multi-layered security approach with encryption and authentication
- **Performance**: Optimized for fast loading times and responsive user interactions

2.2 Product Functions

The system provides comprehensive functionality across the rental lifecycle:

2.2.1 User Management Functions

- **Account Registration**: Email-based registration with email verification
- **Authentication**: Secure login/logout with session management
- **Profile Management**: Comprehensive user profiles with contact information, preferences, and verification status
- **Account Security**: Password reset, two-factor authentication options

2.2.2 Item Management Functions

- **Item Publishing**: Detailed item listings with multiple images, descriptions, pricing, and availability calendars
- **Inventory Management**: Item status tracking, availability updates, and listing modifications
- **Category Organization**: Hierarchical categorization system for improved discoverability
- **Image Management**: Multiple image uploads with automatic optimization and storage

2.2.3 Discovery and Search Functions

- **Advanced Search**: Keyword-based search with multiple filter criteria
- **Geographic Filtering**: Location-based item discovery with distance calculations
- **Category Browsing**: Intuitive category navigation and filtering
- **Recommendation Engine**: Personalized item suggestions based on user behavior and preferences

2.2.4 Booking and Transaction Functions

- **Booking Requests**: Streamlined booking workflow with availability checking
- **Rental Management**: Comprehensive rental history and status tracking
- **Calendar Integration**: Availability calendar with booking management
- **Transaction History**: Complete audit trail of all rental activities

2.2.5 Communication Functions

- **Real-time Chat**: Secure messaging system for rental negotiations and support
- **Notification System**: Email and in-app notifications for booking updates and messages
- **Message History**: Persistent chat history with search capabilities
- **Automated Messaging**: System-generated notifications for key rental milestones

2.3 User Classes and Characteristics

2.3.1 Guest Users (Unregistered)

Characteristics:

- First-time visitors exploring the platform
- Privacy-conscious users researching before registration
- Limited technical expertise expected

Capabilities:

- Browse public item listings
- View item details and images
- Access help documentation and FAQ
- Register for full platform access

Limitations:

- Cannot contact item owners
- Cannot make booking requests
- No access to personalized features

2.3.2 Registered Users (Standard)

Characteristics:

- Active platform participants with verified email addresses
- Varying technical skill levels from beginner to advanced
- Primary revenue generators for the platform

Capabilities:

- Full access to all platform features
- Publish and manage item listings
- Make and manage booking requests
- Engage in real-time chat communications
- Customize profiles and preferences

Subcategories:

- **Item Owners**: Focus on publishing and managing rental listings
- **Renters**: Primarily search for and book available items
- **Hybrid Users**: Both rent and list items on the platform

2.3.3 Platform Administrators (Future Enhancement)

Characteristics:

- Technical staff responsible for platform maintenance
- Customer support representatives
- Compliance and safety monitoring personnel

Capabilities:

- User account management and moderation
- Item listing review and approval
- Platform analytics and reporting
- System configuration and maintenance

2.4 Operating Environment

2.4.1 Client-Side Environment

Supported Browsers:

- Google Chrome 90+ (primary target)
- Mozilla Firefox 88+

- Microsoft Edge 90+
- Safari 14+ (macOS/iOS)

Operating Systems:

- Windows 10/11
- macOS 10.15+
- Linux (Ubuntu 18.04+)
- Mobile: iOS 14+, Android 10+

Hardware Requirements:

- Minimum 2GB RAM
- Modern processor (Intel i3 equivalent or better)
- Stable internet connection (minimum 1 Mbps)
- Display resolution: 320px minimum width (mobile responsive)

2.4.2 Server-Side Environment

Backend Infrastructure:

- Appwrite Cloud or self-hosted instance
- Database: Built-in Appwrite database service
- File Storage: Appwrite storage with CDN integration
- Authentication: Appwrite Auth service

Performance Requirements:

- 99.9% uptime availability
- Sub-2 second page load times
- Concurrent user support: 10,000+ users
- Data backup: Daily automated backups with 30-day retention

2.5 Design and Implementation Constraints

2.5.1 Technology Constraints

Mandatory Technologies:

- **Frontend Framework**: Next.js with React (latest stable version)
- **Backend Service**: Appwrite BaaS for all backend operations
- **Styling Framework**: Tailwind CSS for responsive design
- **Programming Language**: TypeScript for type safety and maintainability

Rationale: These constraints ensure consistency, leverage team expertise, and provide long-term maintainability.

2.5.2 Security Constraints

Data Protection:

- All client-server communications must use HTTPS encryption
- User passwords must be hashed using industry-standard algorithms
- Sensitive data must be encrypted at rest
- Compliance with GDPR and regional data protection laws

Authentication:

- Multi-factor authentication support
- Session management with automatic timeout
- Rate limiting for API endpoints
- Input validation and sanitization

2.5.3 Performance Constraints

Response Time:

- Page load times: < 2 seconds for initial load
- API response times: < 500ms for standard operations
- Real-time chat: < 1 second message delivery
- Image loading: Progressive loading with optimization

Scalability:

- Support for 100,000+ registered users
- 10,000+ concurrent active users
- 1M+ item listings
- Database query optimization for large datasets

2.6 User Documentation

2.6.1 End-User Documentation

Interactive Help System:

- Context-sensitive help tooltips and guides
- Step-by-step onboarding wizard for new users
- Video tutorials for key platform features
- Searchable knowledge base with FAQ

User Guides:

- Getting Started guide for new users
- Best Practices guide for item owners
- Safety and Security guidelines
- Troubleshooting and problem resolution guides

2.6.2 Technical Documentation

Developer Documentation:

- API documentation with examples
- Database schema and relationship diagrams
- Deployment and configuration guides
- Code style guides and contribution guidelines

2.7 Assumptions and Dependencies

2.7.1 User Assumptions

- Users have access to reliable internet connectivity
- Users possess basic computer literacy and web browsing skills
- Users will provide accurate and truthful information during registration
- Users will comply with platform terms of service and community guidelines

2.7.2 Technical Dependencies

External Services:

- Appwrite backend service availability and reliability
- Third-party CDN services for image and content delivery
- Email service provider for notifications and communications
- Domain name system (DNS) and SSL certificate services

Infrastructure Dependencies:

- Cloud hosting provider uptime and performance
- Database backup and recovery services
- Monitoring and alerting systems
- Security scanning and vulnerability assessment tools

2.7.3 Business Assumptions

- Sufficient user adoption to create network effects
- Regulatory environment remains favorable for peer-to-peer platforms
- Market demand for sharing economy services continues to grow
- Competitive landscape allows for differentiation and market entry

---

3. External Interface Requirements

3.1 User Interfaces

3.1.1 General UI Requirements

Design Principles:

- **Responsive Design**: Seamless experience across desktop, tablet, and mobile devices
- **Accessibility**: WCAG 2.1 AA compliance for users with disabilities
- **Intuitive Navigation**: Clear information architecture with minimal cognitive load
- **Visual Consistency**: Cohesive design language with consistent typography, colors, and spacing

UI-1: The system shall provide a responsive web interface that adapts to screen sizes from 320px to 1920px width.

3.1.2 Core User Interface Components

Registration and Authentication Pages

- **Login Page**: Clean, secure login form with "Remember Me" and "Forgot Password" options
- **Registration Page**: Multi-step registration process with email verification

- **Password Reset**: Secure password reset workflow with email verification

Main Application Pages

- **Dashboard/Home Page**: Personalized landing page with recent activity and recommendations
- **Profile Management Page**: Comprehensive profile editing with image upload and verification options
- **Item Listing Management**: Intuitive item creation and editing interface with drag-and-drop image upload
- **Item Discovery Page**: Advanced search and filter interface with map integration
- **Item Detail Page**: Rich media display with booking calendar and owner communication tools
- **Booking Management Page**: Comprehensive booking history and status management
- **Chat Interface**: Real-time messaging with chat history and file sharing capabilities

Navigation Components

- **Sidebar Navigation**: Collapsible sidebar with icons and labels for all major sections
- **Header Navigation**: User account menu, notifications, and search functionality
- **Breadcrumb Navigation**: Clear page hierarchy indication for deep navigation

3.1.3 Mobile-Specific UI Requirements

UI-2: The system shall provide touch-optimized interfaces for mobile devices with minimum 44px touch targets.

UI-3: The system shall support mobile-specific features including camera integration for image capture.

3.2 Hardware Interfaces

IF-1: The system shall operate on standard computing hardware without requiring specialized peripherals.

Hardware Compatibility:

- **Input Devices**: Standard keyboard, mouse, and touchscreen interfaces
- **Display Requirements**: Any device with web browser capability and minimum 320px display width
- **Camera Integration**: Access to device camera for mobile users (optional)
- **Storage**: Local browser storage for offline capabilities and caching

No Special Hardware Requirements: The web-based nature of the application eliminates the need for specialized hardware, ensuring broad accessibility across different device types and economic capabilities.

3.3 Software Interfaces

3.3.1 Backend Service Integration

Appwrite Backend Services:

- **Authentication Service**: User registration, login, session management
- **Database Service**: Data storage, querying, and real-time updates
- **Storage Service**: File upload, management, and CDN delivery
- **Functions Service**: Server-side logic and automated processes

IF-2: The system shall integrate with Appwrite SDK version 12.0 or higher for all backend operations.

3.3.2 Third-Party Service Interfaces

Email Service Integration:

- **Transactional Emails**: Account verification, booking confirmations, password resets
- **Notification Emails**: Booking requests, messages, and system updates

Mapping and Geolocation Services:

- **Location Services**: Address validation and geocoding
- **Map Display**: Interactive maps for item location visualization
- **Distance Calculation**: Proximity-based search and filtering

IF-3: The system shall provide standardized API interfaces for future integration with payment processing services.

3.4 Communications Interfaces

3.4.1 Network Communication Requirements

Security Protocols:

- **HTTPS Encryption**: All client-server communication must use TLS 1.3 or higher
- **WebSocket Security**: Encrypted WebSocket connections for real-time features
- **API Security**: Rate limiting, request validation, and authentication tokens

IF-4: The system shall encrypt all data transmission using industry-standard protocols.

3.4.2 Real-Time Communication

Chat System Communication:

- **WebSocket Protocol**: Real-time bidirectional communication for instant messaging
- **Message Queuing**: Reliable message delivery with offline message storage
- **File Transfer**: Secure file sharing within chat conversations

Notification System:

- **Push Notifications**: Browser-based notifications for critical updates
- **Email Notifications**: Comprehensive email notification system
- **In-App Notifications**: Real-time notification feed within the application

IF-5: The system shall support real-time communication with sub-second latency for chat messages.

4. System Features

4.1 User Authentication and Profile Management

4.1.1 Feature Description

This feature encompasses the complete user lifecycle from initial registration through ongoing profile management. It provides secure, user-friendly authentication mechanisms while enabling rich profile customization that enhances trust and facilitates successful rental transactions.

4.1.2 Stimulus/Response Sequences

Registration Flow:

1. **Stimulus**: User accesses registration page and submits registration form
2. **Response**: System validates data, creates account, sends verification email
3. **Stimulus**: User clicks verification link in email
4. **Response**: System activates account and redirects to login page

Authentication Flow:

1. **Stimulus**: User submits login credentials
2. **Response**: System validates credentials and establishes secure session
3. **Stimulus**: User attempts to access protected resource
4. **Response**: System verifies session and grants/denies access

4.1.3 Functional Requirements

FR-1: The system shall allow users to register using email address and password with the following validations:

- Email format validation and uniqueness checking
- Password strength requirements (minimum 8 characters, mixed case, numbers)
- Required fields: first name, last name, email, password
- Optional fields: phone number, profile picture, bio

FR-2: The system shall authenticate users and manage sessions with the following capabilities:

- Secure login with email/password combination
- "Remember Me" functionality for extended sessions
- Automatic session expiration after 24 hours of inactivity
- Concurrent session management across multiple devices

FR-3: The system shall provide comprehensive profile management including:

- Profile information editing (name, contact details, preferences)
- Profile picture upload and management
- Account settings (password change, email notifications)
- Account deactivation option

FR-4: The system shall implement email verification for new accounts:

- Automated verification email upon registration
- Account activation only after email verification
- Resend verification email functionality

4.1.4 Priority and Dependencies

- **Priority**: High (Critical for platform operation)
- **Dependencies**: Appwrite authentication service, email service provider

4.2 Item Publishing and Browsing

4.2.1 Feature Description

This feature enables the core marketplace functionality by allowing users to publish detailed item listings and providing powerful discovery tools for potential renters. The system supports rich media content, flexible pricing models, and comprehensive search capabilities.

4.2.2 Stimulus/Response Sequences

Item Publishing Flow:

1. **Stimulus**: Registered user initiates item publishing process

2. **Response**: System presents item creation form with validation
3. **Stimulus**: User submits complete item information with images
4. **Response**: System validates, processes images, and publishes item

Item Discovery Flow:

1. **Stimulus**: User enters search criteria or browses categories
2. **Response**: System queries database and returns relevant results
3. **Stimulus**: User applies filters or refines search
4. **Response**: System updates results in real-time

4.2.3 Functional Requirements

FR-5: The system shall allow registered users to publish items with comprehensive details:

- Required fields: title, description, category, daily rental price
- Optional fields: weekly/monthly pricing, deposit requirements, pickup instructions
- Multiple image upload (minimum 1, maximum 10 images per item)
- Availability calendar with blocked dates and custom pricing

FR-6: The system shall provide advanced search and filtering capabilities:

- Keyword search across item titles and descriptions
- Category-based filtering with hierarchical categories
- Location-based search with distance radius options
- Price range filtering and sorting options
- Availability date filtering

FR-7: The system shall display items in an intuitive browsing interface:

- Grid and list view options for search results
- Item preview cards with key information and primary image
- Detailed item pages with image galleries and full descriptions
- Related items and recommendations

FR-8: The system shall support item management for owners:

- Edit item details and update availability
- Pause/unpause listings temporarily
- View item performance metrics and booking history
- Delete items with appropriate confirmation

4.2.4 Priority and Dependencies

- **Priority**: High (Core marketplace functionality)
- **Dependencies**: Appwrite database and storage services, image optimization service

4.3 Booking System

4.3.1 Feature Description

The booking system facilitates the complete rental transaction process from initial request through rental completion. It includes availability management, booking confirmation workflows, and comprehensive tracking for both renters and item owners.

4.3.2 Stimulus/Response Sequences

Booking Request Flow:

1. **Stimulus**: User selects dates and submits booking request
2. **Response**: System checks availability and creates pending booking
3. **Stimulus**: Item owner receives notification and responds to request
4. **Response**: System updates booking status and notifies all parties

Booking Management Flow:

1. **Stimulus**: User accesses booking management interface
2. **Response**: System displays current and historical bookings with status
3. **Stimulus**: User performs booking action (cancel, modify, complete)
4. **Response**: System processes request and updates all relevant parties

4.3.3 Functional Requirements

FR-9: The system shall enable users to request bookings with the following process:

- Date selection with real-time availability checking
- Booking request with optional message to item owner
- Automatic calculation of total rental cost including fees
- Booking confirmation email to both parties

FR-10: The system shall provide booking management capabilities:

- Real-time booking status updates (pending, confirmed, active, completed, cancelled)

- Booking modification requests for date changes
- Cancellation functionality with appropriate policies
- Booking history with search and filtering options

FR-11: The system shall notify relevant parties of booking events:

- Instant notifications for new booking requests
- Confirmation notifications for approved bookings
- Reminder notifications for upcoming rentals
- Completion notifications for rental conclusion

FR-12: The system shall implement booking validation and conflict resolution:

- Double-booking prevention with real-time availability updates
- Booking request timeout for unresponded requests
- Automatic booking status updates based on rental dates
- Booking conflict detection and resolution workflows

4.3.4 Priority and Dependencies

- **Priority**: High (Essential for platform revenue)
- **Dependencies**: Authentication system, item management system, notification system

4.4 Chat System

4.4.1 Feature Description

The integrated chat system provides secure, real-time communication between users to facilitate rental negotiations, coordinate pickup/delivery, and provide ongoing support throughout the rental process. The system maintains conversation history and supports multimedia sharing.

4.4.2 Stimulus/Response Sequences

Message Exchange Flow:

1. **Stimulus**: User composes and sends message in chat interface
2. **Response**: System delivers message instantly to recipient with delivery confirmation
3. **Stimulus**: Recipient views message and optionally responds
4. **Response**: System maintains conversation thread and updates read status

Chat Initiation Flow:

1. **Stimulus**: User initiates chat from item listing or booking page
2. **Response**: System creates new conversation thread or accesses existing one
3. **Stimulus**: System displays chat interface with conversation history
4. **Response**: Users can immediately begin communication

4.4.3 Functional Requirements

FR-13: The system shall provide real-time chat functionality with:

- Instant message delivery and receipt confirmation
- Message history persistence with search capabilities
- Online/offline status indicators for users
- Typing indicators for active conversations

FR-14: The system shall support rich messaging features:

- Text messages with emoji support and formatting
- Image and file sharing with size limitations (max 10MB per file)
- Message editing and deletion with audit trail
- Message reactions and threading for organized conversations

FR-15: The system shall implement chat security and privacy:

- End-to-end encryption for all messages
- User blocking and reporting functionality
- Message content filtering for inappropriate content
- Chat history export for user records

FR-16: The system shall integrate chat with other platform features:

- Booking-specific chat threads with context
- Quick actions for booking-related activities
- Automated system messages for booking status updates
- Chat notifications integrated with platform notification system

4.4.4 Priority and Dependencies

- **Priority**: Medium (Important for user experience)
- **Dependencies**: Real-time communication infrastructure, user authentication, content moderation system

4.5 Responsive Navigation

4.5.1 Feature Description

The navigation system provides intuitive, consistent access to all platform features across different devices and screen sizes. It includes a collapsible sidebar for desktop users and mobile-optimized navigation patterns for smaller screens.

4.5.2 Stimulus/Response Sequences

Navigation Flow:

1. **Stimulus**: User clicks on navigation menu item or link
2. **Response**: System navigates to requested page with loading indicator
3. **Stimulus**: Page loads successfully
4. **Response**: System updates navigation state and displays requested content

Mobile Navigation Flow:

1. **Stimulus**: User taps mobile menu toggle
2. **Response**: System expands/collapses mobile navigation menu
3. **Stimulus**: User selects navigation option
4. **Response**: System navigates to page and closes mobile menu

4.5.3 Functional Requirements

FR-17: The system shall provide comprehensive navigation structure:

- Sidebar navigation with icons and labels for all major sections
- Breadcrumb navigation for page hierarchy understanding
- Quick access navigation for frequent actions
- Search functionality accessible from all pages

FR-18: The system shall implement responsive navigation patterns:

- Collapsible sidebar for desktop and tablet views
- Mobile-first navigation with hamburger menu
- Touch-optimized navigation elements for mobile devices
- Consistent navigation behavior across all device types

FR-19: The system shall support navigation accessibility:

- Keyboard navigation support for all navigation elements
- Screen reader compatibility with proper ARIA labels
- High contrast navigation options for visual accessibility
- Focus indicators for navigation elements

FR-20: The system shall provide navigation state management:

- Active page highlighting in navigation menu
- Navigation history for browser back/forward functionality
- Deep linking support for all major pages
- Navigation loading states and error handling

4.5.4 Priority and Dependencies

- **Priority**: Medium (Important for usability)
- **Dependencies**: Frontend framework, responsive design system

---

5. Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Response Time Requirements

NFR-1: Page Load Performance

- Initial page load: ≤ 2 seconds for first-time visitors
- Subsequent page loads: ≤ 1 second for cached content
- API response time: ≤ 500ms for standard database queries
- Search results: ≤ 1 second for filtered item searches

NFR-2: Real-Time Communication Performance

- Chat message delivery: ≤ 1 second end-to-end latency
- Notification delivery: ≤ 2 seconds for in-app notifications
- Live updates: ≤ 3 seconds for booking status changes
- File upload progress: Real-time progress indication for uploads > 1MB

5.1.2 Throughput Requirements

NFR-3: Concurrent User Support

- Support minimum 1,000 concurrent active users
- Handle 10,000+ page views per hour during peak usage
- Process 500+ simultaneous booking requests
- Maintain performance during 5x traffic spikes

NFR-4: Data Processing Requirements

- Image processing: ≤ 10 seconds for image optimization and thumbnail generation
- Database queries: Support 10,000+ queries per minute
- Search indexing: Real-time search index updates within 30 seconds
- Backup processes: Complete daily backup within 4-hour window

5.2 Safety Requirements

5.2.1 Data Protection and Recovery

NFR-5: Data Backup and Recovery

- Automated daily database backups with 30-day retention
- Point-in-time recovery capability within 1-hour RPO (Recovery Point Objective)
- Disaster recovery with 4-hour RTO (Recovery Time Objective)
- Geo-redundant backup storage for critical data protection

NFR-6: Input Validation and Sanitization

- Server-side validation for all user inputs
- SQL injection prevention through parameterized queries
- Cross-site scripting (XSS) prevention with input sanitization
- File upload validation with virus scanning and content verification

5.2.2 Error Handling and System Stability

NFR-7: Graceful Error Handling

- User-friendly error messages without technical details
- Automatic error reporting and logging for system monitoring
- Fallback mechanisms for third-party service failures
- Graceful degradation of features during partial system outages

5.3 Security Requirements

5.3.1 Authentication and Authorization

NFR-8: User Authentication Security

- Multi-factor authentication option for enhanced security
- Password hashing using bcrypt or Argon2 with appropriate salt rounds
- Session management with secure, httpOnly cookies
- Account lockout after 5 consecutive failed login attempts

NFR-9: Data Encryption and Protection

- All data transmission encrypted using TLS 1.3 or higher
- Sensitive data encrypted at rest using AES-256 encryption
- Personal information anonymization in logs and analytics
- Secure key management with regular key rotation

5.3.2 Privacy and Compliance

NFR-10: Privacy Protection

- GDPR compliance with user consent management
- Data minimization principle - collect only necessary information
- Right to data portability and deletion upon user request
- Privacy-by-design approach in all feature development

NFR-11: Security Monitoring and Auditing

- Comprehensive security event logging
- Real-time monitoring for suspicious activities
- Regular security vulnerability assessments
- Incident response plan with defined escalation procedures

5.4 Software Quality Attributes

5.4.1 Usability Requirements

NFR-12: User Experience Standards

- Intuitive interface requiring minimal learning curve for basic operations
- Maximum 3 clicks to reach any major platform feature
- Consistent design patterns across all pages and features
- Mobile-first responsive design with touch-optimized interfaces

NFR-13: Accessibility Standards

- WCAG 2.1 AA compliance for web accessibility
- Screen reader compatibility with proper semantic markup
- Keyboard navigation support for all interactive elements
- Color contrast ratio compliance for visual accessibility

5.4.2 Reliability and Availability

NFR-14: System Availability

- 99.9% uptime availability (maximum 8.77 hours downtime per year)
- Planned maintenance during low-traffic hours with advance notification
- Automatic failover capabilities for critical system components
- Load balancing to distribute traffic and prevent single points of failure

NFR-15: System Reliability

- Mean Time Between Failures (MTBF) of 720 hours for critical functions
- Automatic system health monitoring with proactive alerting
- Database transaction integrity with ACID compliance
- Fault tolerance with automatic recovery for transient failures

5.4.3 Scalability and Maintainability

NFR-16: Scalability Requirements

- Horizontal scaling capability to accommodate user growth
- Database partitioning strategy for large datasets
- CDN integration for global content delivery
- Microservices-ready architecture for future service decomposition

NFR-17: Maintainability Standards

- Modular codebase with clear separation of concerns
- Comprehensive code documentation and inline comments
- Automated testing coverage minimum 80% for critical functions
- Version control with branching strategy for release management

5.5 Business Rules

5.5.1 User Access Rules

NFR-18: Registration and Access Control

- Only registered users with verified email addresses can publish items
- Only registered users can initiate booking requests
- Guest users limited to browsing public item listings
- Account suspension capability for terms of service violations

NFR-19: Transaction Rules

- Mandatory acceptance of terms and conditions before first booking or listing
- Minimum user age requirement of 18 years for platform participation
- Geographic restrictions based on local regulations and service availability
- Anti-fraud measures with automated suspicious activity detection

5.5.2 Content and Listing Rules

NFR-20: Content Moderation

- Prohibited item categories clearly defined and enforced
- Automated content filtering for inappropriate or illegal items
- User reporting system for community-based moderation
- Manual review process for flagged content with defined response times

6. Other Requirements

- **GDPR and Data Protection Compliance:** The system must comply with GDPR and other relevant data protection regulations including user consent management, data portability, and right to deletion.
- **WCAG Accessibility Standards:** The system must meet WCAG 2.1 AA accessibility standards to ensure usability for users with disabilities, including screen reader compatibility and keyboard navigation.
- **Content Moderation and Safety:** Implementation of prohibited item categories, automated content filtering, and user reporting system for community-based moderation of inappropriate content.
- **Payment Gateway Integration Readiness:** System architecture must be prepared for future integration with payment processing services (Stripe, PayPal) with secure API endpoints and transaction audit trails.
- **Data Backup and Recovery:** Daily automated database backups with 30-day retention, disaster recovery procedures, and point-in-time recovery capabilities to ensure business continuity.

Appendices

Appendix A: Glossary

- **API (Application Programming Interface)**: A set of protocols and tools for building software applications, specifying how software components should interact.
- **Appwrite**: An open-source Backend-as-a-Service (BaaS) platform that provides developers with all the core APIs required to build any application.
- **Booking**: The process of requesting and confirming the rental of an item for specific dates, including payment processing and terms agreement.
- **BaaS (Backend-as-a-Service)**: A cloud service model that provides developers with a way to link their web and mobile applications to backend cloud storage and APIs.
- **CDN (Content Delivery Network)**: A geographically distributed network of proxy servers and their data centers that provide high availability and performance by distributing the service spatially relative to end-users.
- **Chat**: Real-time messaging functionality that enables direct communication between platform users for rental coordination and support.
- **GDPR (General Data Protection Regulation)**: A European Union regulation on data protection and privacy for all individuals within the EU and European Economic Area.
- **Next.js**: A React-based web development framework that enables functionality such as server-side rendering and generating static websites.
- **P2P (Peer-to-Peer)**: A decentralized communications model where each party has the same capabilities and either party can initiate a communication session.
- **React**: An open-source JavaScript library for building user interfaces, particularly web applications with complex, interactive user interfaces.
- **SRS (Software Requirements Specification)**: A document that describes the intended purpose, requirements, and nature of a software system.
- **TLS (Transport Layer Security)**: A cryptographic protocol designed to provide communications security over a computer network.
- **UI/UX (User Interface/User Experience)**: UI refers to the visual elements users interact with, while UX encompasses the overall experience and satisfaction users have when interacting with the system.
- **WCAG (Web Content Accessibility Guidelines)**: A set of recommendations for making web content more accessible to people with disabilities.

Appendix B: Analysis Models

B.1 Use Case Diagrams

To be developed during design phase

Primary Use Cases:

- User Registration and Authentication
- Item Publishing and Management

- Item Search and Discovery
- Booking Request and Management
- Real-time Chat Communication
- Profile Management

B.2 Sequence Diagrams

To be developed during design phase

Key Sequences:

- User Registration Flow
- Item Publishing Process
- Booking Request Workflow
- Chat Message Exchange
- Payment Processing Integration (Future)

B.3 Data Flow Diagrams

To be developed during design phase

Data Flow Levels:

- Level 0: System Context Diagram
- Level 1: Major Process Decomposition
- Level 2: Detailed Process Flows

B.4 Entity Relationship Diagrams

To be developed during design phase

Core Entities:

- Users (Renters and Owners)
- Items (Rental Listings)
- Bookings (Rental Transactions)
- Messages (Chat Communications)
- Categories (Item Classifications)

Appendix C: To Be Determined List

Currently, there are no major unresolved requirements or design decisions that require further clarification. All critical aspects of the system have been defined to enable development to proceed.

Future Considerations:

- Payment processing service selection and integration details
- Advanced analytics and reporting requirements
- Mobile application development specifications
- Scalability testing and performance benchmarking criteria
- Third-party integration priorities and timelines